Release 1.0

90-6 30 June 1990

ON BEYOND WINDOWS: DYNABOOK REVISITED

Microsoft announced Windows 3 with more hoopla than any computer event since the NeXT announcement in 1988. But Windows is a predictable -- and long-awaited -- advance. It lets the PC catch up with the Mac and swing ahead: It gives the PC the features of a Mac, but it has a potential installed base starting at 30 million vs. the Mac's 5 million.

Yet revolutions (unlike earthquakes) don't happen twice the same way. Windows represents an upgrade to the current installed base and has been constrained so it can run existing programs. Even the software developed specifically for it, with a few exceptions, is the same old stuff prettied up. You could say the same for the NeXT machine, or the PS/1 (page 12).

So what will it take to support genuine advances in applications and attract a fundamentally new customer base? A fundamentally new platform supporting fundamentally new applications. The next revolution in personal computers will come from outside the establishment.

Our guess is that it will happen on those nifty little portable machines -- variously called notepad, pen-based or handwriting-recognition -- that are getting so much attention right now. The killer application leading the way, we believe, will be calendar/communications management; it's the only widely performed yet specific task still left to automate effectively.

Here's our rationale for this system -- call it the HandyWidget -- followed by brief profiles of some early players (page 15). Aside from the GRiDPAD, few products have been announced, but companies in the field include GO, SuperScript, Communication Intelligence, Momenta and Slate.

HandyWidget is not a code name for any machine now in development; it's our vision of what any of them could be. It's speculation, not inside information. Certainly, not all of HandyWidget's features and functionality will appear in version 1.0 of anything on the horizon. Version 1 will probably be a simple personal mail/fax tool with a calendar interface/metaphor; automatic handling of communications won't come until later. ⟹

JULY -- MAKING MARKETS

INSIDE

ON BEYOND WINDOWS: DYNABOOK 1
How to make a revolution.
The HandyWidget user. 4
The business case.
Multiplying technologies.
Natural language.
Box: The PS/1 is NeXT? 12
WOULD-BE REVOLUTIONARIES 15
GO, SuperScript, Momenta, Slate, et al.
PHONES/RESOURCES/CALENDAR 19-23
Names, phones, books, dates.

The personal computing business began with typists and writers, who used dedicated word-processors from Wang and IBM without realizing they were *computers*. The Apple II and VisiCalc got the early techy financial analysts, the PC and 1-2-3 attracted a broad range of numbers-oriented business people, the Mac and PageMaker got the graphic artists and document layout folks. Each platform was used as an application machine for a particular task that worked uniquely well on that particular platform. Only later did each machine/environment broaden its market, once it had a base from which to enter the mainstream.

The notepad computers will address two markets -- the first market of more or less traditional vertical applications, and a new one focusing on personal communications and calendar management. In each case, it will be the applications, not the platforms per se, that sell. On the hardware side, this market will probably belong to such long-lived players as GRiD and a large number of Japanese vendors such as Sony, expert in miniaturization and recognition technologies.

On the software side, things are more interesting. Since you can't just port the same old applications over directly, there's likely to be a profusion of DOS-based development environments, just as there were a variety of alternatives to Windows. But we doubt any of them will gain critical mass as a *development* environment, in the face of a promised offering from Microsoft. Nonetheless, the new hardware platform with a small form factor, digital signal processing/recognition, instant on/off, multi-tasking, quiet operation and solid-state storage will require new systems software -- and will open the market for a wholly new platform rather than an extension of DOS/Windows. And we believe someone, probably GO, has a chance of using that opening to create a new standard for a *user* environment.

All the hoopla has focused on handwriting recognition, to the point that many people seem to think that that's all there is. But it's the underlying software paradigms that make the difference.

DOS is for the developer; the developer's application is for the user

The "traditional" market of vertical applications is easy to discern: relatively structured applications, where (say) a builder-user will create a general application for use by salespeople or agents. They may be taking orders or recording inventory into a form, assessing accident damages by marking up diagrams (one set of diagrams for each brand of car), repairing equipment with the aid of on-line documentation including images and expert-system diagnostics, following a checklist as they inspect a property for a mortgage -- any task that can be made routine by programming the procedures and help into the system.

DOS-based tools will contend with special-purpose hardware/software environments such as SuperScript's (page 16). In most cases, DOS is likely to win. Since the builder-user is planning to build a single (if flexible) application, he'll do whatever it takes without worrying about a standard platform or its richness. He can build his own world on top of DOS and spread the cost of doing so (he hopes) over a large number of users. His environment is a tool used to develop the particular application on each user's system

-- but the value of what he produces will lie in the application, not in the tool. At the supplier's option, the user may also have access to other applications or HandyWidget-class functions. But overall the builder-user who develops the application wants the user to concentrate on the job at hand.

A variety of vendors will probably provide development tools on top of DOS or Windows with handwriting-recognition technology embedded, and a stylus instead of a mouse-driver, creating layered versions of the new development environments much as Windows creates a layered version of the Mac. Others will supply drawing tools, forms packages, database links, communications tools and handwriting-, character-, and voice-recognition technology. This market is also an excellent target for Softview's If:x technology for forms that dynamically resize and rearrange themselves as data is entered. (Don't try writing smaller on one of these systems to fit more data in -- the text it reads is generally converted to a standard size, although we can imagine a layout tool that would flout this convention.)

For example, an application could recognize defined gestures and symbols for insurance adjusters: a circle around a component means it's damaged, and a scale pops up for the user to indicate amount of damage graphically; an X through it means it's totally destroyed. (Back home, a host system will take this information, determine the cost of replacement parts and add in hours of labor times the prevailing labor rate in the car-owner's community.

The resulting vertical applications will proliferate, but to the user they will be single-purpose machines, self-contained within their vertical niche, not tools like, say, a spreadsheet, which is a modeling tool. The end-user will rarely see the full power of the platform, but just one sample of what it can be programmed to do. Thus these applications will establish a new style of computing, but they're unlikely to help establish a single new platform (although that's the goal of SuperScript). That's the role of...

The HandyWidget: Not just a pretty interface

Let's get to the point -- the system that we think will change the world. The major, initial customer for HandyWidget is the life-is-never-routine executive on the go (pages 4 and 5). This user will interact directly with the new environment, not just with applications created in it.

People who talk about handwriting recognition as the key to the next generation of computers miss the point. Yes, they won't have keyboards except as an option. And yes, you do use a pen (to be fancy, a stylus) for input. But the stylus won't *replace* the keyboard. The HandyWidget will do more -- and less -- than full handwriting recognition. More, because it will also understand gestures -- rub-out, circle, move down and so forth. Less, because it won't do well recognizing cursive handwriting (most people don't do a good job writing it!). Besides, writing on a screen is tiring. It's just not a good way to enter large amounts of data.

What this means is a new style of computing, for a new audience. It's not people who don't type, but people who don't *write*. They may dictate a letter or two, and clarify a sales report or dash off a memo. They may file orders or note the outcome of a meeting or fill in forms, but they don't write.

page 6 ⟹

The HandyWidget user: too busy to be technical

Who is the HandyWidget user? She or more likely he is on the go, managing by traveling around. He may be a ceo or a middle manager or a salesperson in a large organization, or a small businessperson trying to get by without a secretary. He probably does use not a pc in the office, although he may have a secretary who does. He rarely writes, let alone types. He spends his time talking to people, and reading and scribbling on memos. He abhors paperwork, and does too much of it. (He would love a system that could reduce the paperwork, which he tries to leave to his secretary.) He sees his world (his home base) not as a form or a spreadsheet or a laid-out document or a pie chart or a desktop where you can leave things scattered in a pleasing but meaningless pattern...but as a calendar.

The resource he monitors most carefully is his time and the people and events that fill it. The way he organizes things is by time: First the breakfast with Juan, then the performance review with Alice, the phone calls to three top prospects, the visit to the dentist, the planning meeting over lunch with Videoton, the briefing with the sales team, the letter to Mr. Arkady, the articles to read, the quality time with the kids....all these are filed in some order under "TODAY." The performance review with Alice has links to the personnel file; if he had a different kind of meeting with Alice -- a strategy session, say -- there might be links to the marketing plan for her territory. He sees the world as objects filtered through and ordered by his schedule.

The system this person needs is not a spreadsheet or word-processor or graphical charting package; it's an electronic calendar and a mailbox the way a spreadsheet is an electronic financial table. This calendar is not a mere listing of events that he can enter or erase, a set of boxes displayed one day at a time in 30-minute increments. It's an active calendar linked to the outside world, with items prioritized, cross-referenced and sequenced. The restaurant for the lunch meeting is an object, with an address and opening hours attached; if the woman from Videoton wants to know how to find it, he could easily fax her the map by selecting the "confirm" button. Now all he has to do is select options or use the defaults; options include fax, E-mail or local printer for snail mail; and map or directions.

The active calendar

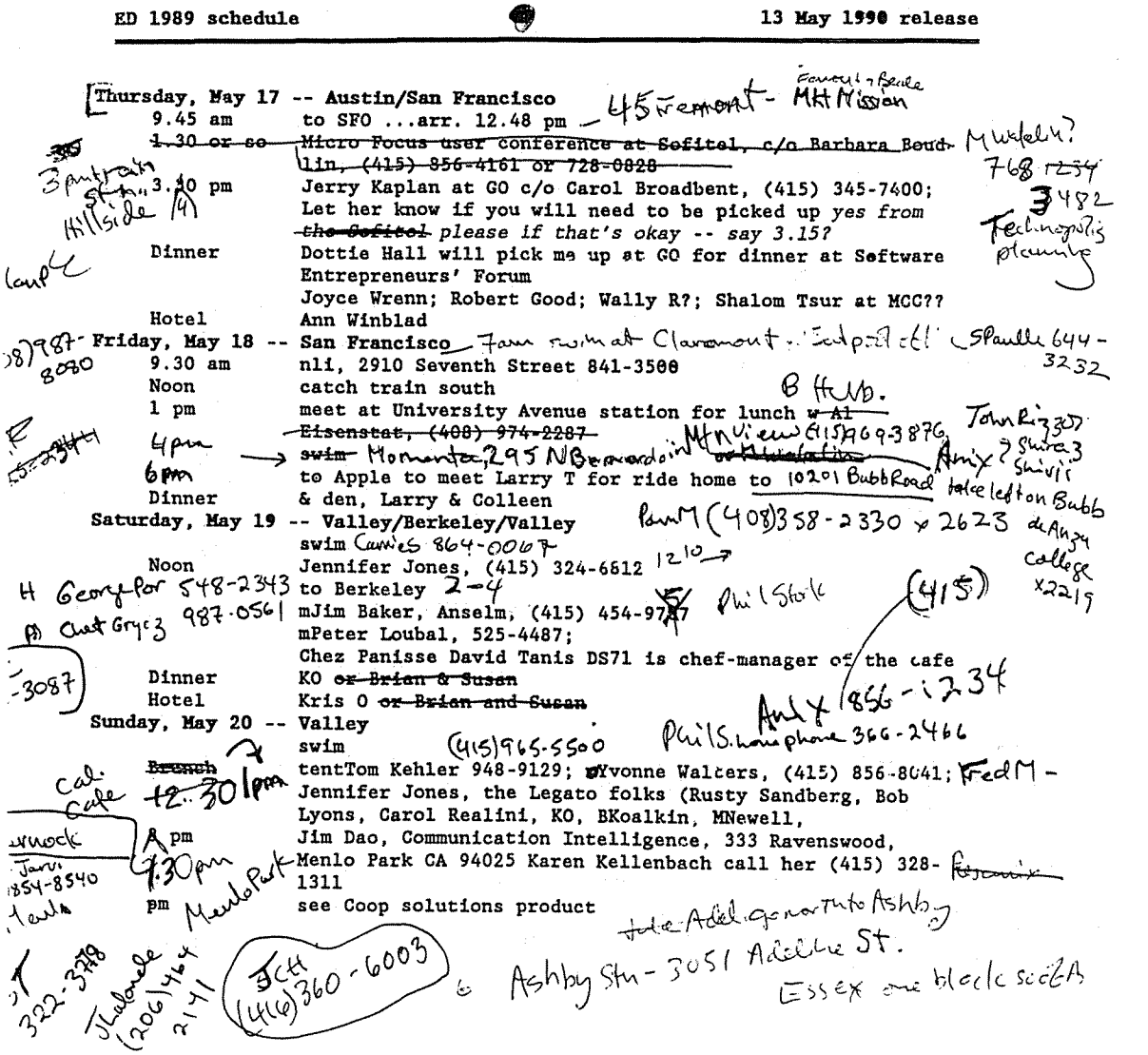
The user can program his schedule, manipulate it, reorganize it. He can move appointments around according to assigned priorities, call up data on the individuals he's meeting, enter memos about the meetings and post them to the right files automatically. He can schedule a follow-up meeting, thereby generating a message requesting confirmation. He can create follow-up letters with appropriate comments ("Thanks for the order," "I look forward to working with you," "I'm sorry we can't meet your needs this time") by checking a box or two. He can answer a letter by writing in "Tell him no!" or by underlining the option you prefer, adding "by next Weds latest." He can mark up a speech draft assembled from boilerplate and have HandyWidget produce a finished copy.

HandyWidget also provides access to outside applications. It can talk to word-processors and databases, accounting systems and home-office order-entry applications. Ideally, it's hooked into Motorola's worldwide cellular

network; at first, it will just use a modem and maybe a scanner. It will have a calendar-programming language that "understands" the user in terms of appointments, follow-ups, schedule conflicts, locations, travel distances and times, and their relationships -- and talks to other applications in terms of objects, files, database tables, application program interfaces. It will also understand many of the routine messages the user receives: Juan called about Alice, Fred called to cancel, Mr. Wunsch called to reschedule to next week.

Type not, write not

Take a look at the calendar page below. The amount of new data written in was minimal, and much of it was redundant (or inconsistent; see top left!) because we don't yet have a HandyWidget. Most of the information it was in a computer already (including databases, memos, documents, e-mail). Someone else could have done all the typing; HandyWidget just needs to be able to get at that information in a way that's meaningful. It's the linking of people, times & dates and places that contains the user's new information. Imagine the convenience of letting the HandyWidget do the rearranging and linking to existing information automatically!



ED 1989 schedule **13 May 1990 release**

Thursday, May 17 -- Austin/San Francisco
9.45 am to SFO ...arr. 12.48 pm
~~1.30 or so~~ ~~Micro Focus user conference at Sofitel, c/o Barbara Boudlin, (415) 856-4161 or 728-0828~~
3.40 pm Jerry Kaplan at GO c/o Carol Broadbent, (415) 345-7400; Let her know if you will need to be picked up *yes from ~~the Sofitel~~ please if that's okay -- say 3.15?*
Dinner Dottie Hall will pick me up at GO for dinner at Software Entrepreneurs' Forum
Joyce Wrenn; Robert Good; Wally R?; Shalom Tsur at MCC??
Hotel Ann Winblad

Friday, May 18 -- San Francisco
9.30 am nli, 2910 Seventh Street 841-3500
Noon catch train south
1 pm meet at University Avenue station for lunch ~~w Al Eisenstat, (408) 974-2287~~
4pm ~~swim~~ to Apple to meet Larry T for ride home to
6pm
Dinner & den, Larry & Colleen

Saturday, May 19 -- Valley/Berkeley/Valley
swim
Noon Jennifer Jones, (415) 324-6812 to Berkeley
mJim Baker, Anselm, (415) 454-9787
mPeter Loubal, 525-4487;
Chez Panisse David Tanis DS71 is chef-manager of the cafe
Dinner KO ~~or Brian & Susan~~
Hotel Kris O ~~or Brian and Susan~~

Sunday, May 20 -- Valley
swim
~~Brunch~~ tentTom Kehler 948-9129; mYvonne Walters, (415) 856-8041;
Jennifer Jones, the Legato folks (Rusty Sandberg, Bob Lyons, Carol Realini, KO, BKoalkin, MNewell,
pm Jim Dao, Communication Intelligence, 333 Ravenswood, Menlo Park CA 94025 Karen Kellenbach call her (415) 328-1311
pm see Coop solutions product

The data in(to) the machine

The data will get into the HandyWidget from other sources -- downloading over phone lines or directly from a "base" computer in the home office. The HandyWidget has a modem and can receive faxes; it may also include an OCR scanner option. Or it could download scanned-in, converted-to-text material as E-mail. The HandyWidget will recognize speech and words as snippets of data (objects) in a particular context, so that the system "knows" what it's looking or listening for. Thus it will eventually handle all forms of communication including voice, parsing them and generating responses to most of them according to the user's instructions. (You can see how these capabilities could be useful on a desktop as well.)

Applications will tend to be for annotation rather than creation of data. The data users enter directly will be mostly annotations, changes, mark-ups, check-marks, menu selections, short entries into fields of forms.

Behind the magic

While discussion about notepad computers focuses on the interface, the HandyWidget's real power lies beneath the interface, in capabilities that will make it far more flexible and intelligible than the development environment of the first kind of notepad computer:

- *data included*. The HandyWidget already includes most of the data the user needs, so he doesn't need to type or even write much. It has the necessary communications hardware and tools to receive data from outside -- modem, fax, networking, maybe a cellular phone some day.

- *object-oriented with user objects*. The data is represented as objects, so that it can be found and manipulated with ease. The objects are business items, such as people, meetings, orders, budgets, not screen objects such as circles or system objects such as files. The system will understand the context surrounding these objects and thereby appear to understand the user.

- *high-level language*. The user can give the HandyWidget instructions, such as to send a follow-up letter or change a meeting. The language will be implemented both as written instructions and as selections from a context-sensitive menus, again to minimize data/command entry.

- *interoperability*. The HandyWidget can communicate not just with other hardware and operating system environments, but with other applications and data formats. And of course the HandyWidget's own applications can communicate with each other.

- *coordination and reconciliation*. It knows how to update and coordinate among its own applications and with partner systems across gaps in time and space.

- *recognition capabilities*. And oh, yes, the HandyWidget can understand handwriting and gestures and graphical objects, scanned text (including faxes), and ultimately voice. It has natural-language capabilities to help in these functions and to parse messages.

The business case: You can't cross a chasm a step at a time

The techies will say: "This is an operating system? It's too high-level; it's an application." But that's where software has been moving all along. We're not suggesting that calendar/communications management is the only application that people will use on a notepad computer. Like 1-2-3 and PageMaker, we believe, the calendar manager will establish its platform broadly enough to make it an alluring base for other applications vendors. But unlike spreadsheets or even documents (the basic metaphor for NewWave and Interleaf's Active Documents among others), a calendar is a universal, familiar implement that can encompass most people's range of activities. The calendar is uniquely suited to act as the home base from which people handle a broad range of tasks.

In the long run, HandyWidget's software or at least the important high-level parts of it may well migrate back to PCs or Macs or UNIX boxes. (Or at least the concepts will be picked up, much as Windows 3.0 now effectively offers most of what the Mac does -- and other benefits besides. Consider Apple's plaintive ad: "Now that everyone agrees how a computer should work, try the only one that actually works that way." GO's marketing manager should keep that in her textbase.)

Millions of people consider their personal computer a 1-2-3 machine; millions more feel that same cozy way about their word-processor (for us, it's XyWrite). There's probably an even larger number of millions who would feel that way about a calendar machine, if they didn't have to type or click, but could just write, circle, move, cross out and mark up. That's the promise of the HandyWidget.

Business strategies: Grow the pie

What we're describing isn't inevitable. It's a challenge to establish a new software-hardware standard, as Apple's early struggles with the Mac show, even though it has succeeded for now. And as the cases of, say, Symbolics and Metaphor show, it's even harder to go from being a hardware-based vendor to selling only the software that's your real value-added. The way to start, of course, is to license your software broadly, so that it gets wide support and has a chance of becoming a new standard. It seems a given that the hardware platforms will end up coming mostly from Japan, but GO has a chance at providing the software platform of the future, if it can move it off its own hardware platform to something more universal (page 15).

Behind the magic: Go forth and multiply (don't just add)

Where will we get the power to do all this? Object-oriented programming, natural language, character-recognition and the like all use resources extravagantly. But that's what the 486, RISC chips and cheap memory are for. Most users don't need extra-high-resolution graphics, multivariate statistical analysis or 20-digit precision; they need the intelligibility, flexibility and appearance of understanding that come from symbolic computation.

We are taking the hardware for granted; we assume instant on/off operation, capacious, reliable memory and storage, signal processing for handwriting and ultimately voice input, modem, fax, scanning and OCR capabilities. On top of this there's the software environment, with natural language under-

standing to aid in character and word recognition as well as application-specific context-definition, interoperability both among applications on the system and with foreign environments, object-oriented programming with class libraries that contain user objects (not just screen or system objects), forms/semi-structured messages, E-mail, and hypertext. Note that few of these things are technologically linked to notepad computing per se; they're just all coming to maturity simultaneously.

None of these is really new. We've been writing about products embodying them for years. But these products are half-measures -- fine in their own right but unable to broaden their reach because they operate standalone in a (mostly DOS) world that can't properly support them. They include Lotus Agenda (see **Release 1.0**, 87-10, 88-3); Interleaf's Active Documents (90-3); The Coordinator and Information Lens and their E-mail descendants Oracle-Mail, Agility and Beyond (86-10, 88-2, 89-10); natural language (90-3, 89-7); text management and filtering tools (87-11, 88-1, 88-4, 89-7, 89-12, 90-2, 90-3); Cooperative Solutions (to build defined transactions, 89-6); forms technology such as Softview's If:x; object-oriented programming and databases (87-3, 88-9, 88-12, 89-9); HAL for 1-2-3 (85-9, 86-5); Magellan (89-1); Lotus Notes (88-3, 89-12); NewWave (87-12); remote procedure calls, SQL, dynamic data exchange and Lotus DataLens; and many more. Like all the window managers and other tools on top of DOS, they don't work together.

The promise of the HandyWidget is that the functions these packages comprise can be *multiplied*, not added, together. That is, they must know about each other and about themselves. The user's environment is not a hermetically sealed, internally oriented tool, but rather a home base from which he can talk to and manipulate the capabilities of other applications. The goal HandyWidget is to build an operating environment that either encompasses these capabilities, or that is so compelling it encourages innovative developers to do their software over from scratch to fit in this new environment.

Slate's Vern Raburn recently told PC Week: "Things like SQL, groupware -- they're important, and they'll sell billions, but God, they're boring." Yes, but they're integral to the environment his company is developing for. And indeed they are boring. The achievement of HandyWidget will be to make them ignorable -- just as the achievement of democracy is to allow most people to ignore government. It's not that it goes away, but that it runs effectively and unobtrusively.

The natural language it understands

The original pcs entered a market where the end-user had to type in most of the data, although more recently it has become possible to download data from mainframes. The HandyWidget is entering a different world; it's really a peripheral to existing banks of data -- not necessarily databases -- which will be organized into objects for accessibility and manipulability. So the HandyWidget doesn't need a data-intensive input mechanism such as a keyboard; it needs one that's good at communicating changes and modifications, and that's easy for everyone to use. To provide good voice-recognition (and probably good handwriting-recognition too) will require some rudimentary natural-language understanding. There's simply too much ambiguity in English (e.g., here and hear) for the system to do much without it.

A machine Claude Shannon could love?

Claude Shannon crystallized the world's understanding of information -- that area between randomness and predictability where each bit matters. (For details, see Jeremy Campbell's "Grammatical Man," Simon & Schuster 1982.) Each individual bit may be meaningless; when the bits form patterns, meaning emerges, but other bits become redundant. A huge amount of information can still be fairly useless until it is properly organized. Once it's organized, you have to add very little information to provide a lot more meaning. The more concrete the context, the more meaningful each additional piece of information -- or the more redundant.

To be less abstract, imagine a computer system full of documents, address lists, schedules, accounting data, expense accounts and the like. As long as each set of information is maintained separately, there's going to be lots of possibly inconsistent information in there with lots of unrecognized redundancy. Each time you refer to something, you have to include all the necessary data about it.

But if the system is integrated, and data is kept as objects, then someone's name implicitly includes all the other information relating to that person. Information that already exists needn't be entered again. So information storage will be more efficient, and overall performance will be enhanced. Although it may take a little longer to find each object, you'll find (pointers to) the whole object, and not just the part that corresponds to the application you're in. It will take less work to explain to the system what you want, because it already "knows" so much -- and is capable of learning through the object-stripper. Indeed, the value of object-oriented approaches is not just reusable code -- it's reusable and consistent *data*.

This approach to information underlies everything the HandyWidget can do. We won't go into a long discussion of object-oriented databases as a way to store information, but you can read **Release 1.0**, 89-9, if you'd like details. As we said there, the valuable information isn't so much in the data as in the links and the structure.

Specifically, HandyWidget will need an "object-stripper," a tool that knows how to find objects hidden in seemingly dull files. This is a powerful system that probably won't show up on the HandyWidget right away, but it could work on a host and send objects to HandyWidget. The closest thing to an implementation right now is probably the categorization tool written by Reuters and Carnegie Group (**Release 1.0**, 90-3), which analyzes/parses text to determine topics of news stories. Potentially, the underlying technology, Carnegie's Text Categorization System, could parse all kinds of communications to recognize items of interest to a given organization: topics of complaints, orders, price changes, sales trends (all references to soft-drink sales, say), meetings, events. At least at first, most of the source information would be stored and analyzed at the server; only derived objects and relationships would be kept on the user's system.

Natural language understanding is not a binary concept (see **Release 1.0**, 90-3), nor does it necessarily have much to do with English. It's a set of ca-

pabilities ranging from interpreting a menu selection or classifying an article by keywords or picking out date and time in a message about a meeting, to interpreting plain text. In a data-processing sense, natural-language understanding is the ability to pick out objects from a text and determine the relationships among them, and to construct a context in which to interpret further input. HandyWidget must be powerful enough to identify the appropriate objects to determine a context and move on to finer-grained parsing from there.

For example, Alice is an employee and a salesperson; Juan is a potential customer; "items" and "units" and "29 Grade A" and "the low-end model" are products that Alice sells to Juan and other customers. Once HandyWidget determines that the message is from Juan, it can review his correspondence and understand what he means by "I'd like to go ahead with the purchase," or "Please send me the product specs." (Other phrases are more straightforward, such as "Thank you for your letter.") What's the tangible result of the parse? The system displays a short list of these actions it plans to take (entering a purchase order and assigning the account to Alice, say) next to the letter, with the sentences it understood referenced and those it couldn't figure out highlighted.

HAL in context: A digression

In this context, you might wonder about the virtual disappearance of HAL, a 1-2-3-specific natural language tool (acquired by Lotus along with developers Bill and Larry Gross, who subsequently developed Magellan). The product got somewhat lost in the Release 2.2/3 delays, stopped being promoted and disappeared except for some loyal users who refuse to upgrade without it. It was positioned mostly as an English front-end, and failed to find a market; once people take the trouble to learn an application, they're usually willing to take the trouble to learn the language (menu items, short commands, syntax) that goes with it; there are only so many things the application can do anyway, and it's easier to pick the action off a short menu or type a specific command than to converse with your system. The proper role of *English* (or French or Russian) natural language is probably more to parse mail and input from foreign systems, since the user of a pen machine probably prefers terse communication anyway. (Try quoting poetry at your desk some day and see how many *people* understand you.) Why type it in if you can select a command or an object and save the trouble?

But the true benefit of HAL (and of Ashlar Vellum; **Release 1.0**, 89-12) is not so much *English* understanding as its ability to watch and anticipate what the user is trying to do -- or *application* understanding. With that, it can present the right selection of possible choices, a capability functionally equivalent to disambiguating correctly a user's incoherent input. It understands the "grammar" not just of English but of the application/task the user is working on.

Semi-structured messages and forms

For the traditional form-based applications -- fill in these fields and upload your data to dBASE or SQL Server at the end of the day -- products such as the GriDPAD or other DOS machines are just fine. But there's a richer kind of form that involves objects and semi-structured messages -- where the forms provide a context in which the text can be interpreted.

Fortunately, most messages adhere to some pattern, and tend to refer to objects already in the system.[1] As parsing systems get smarter on the one side, text will become more parsable on the other as people start to communicate more concisely using forms. Within a single organization that's easy, and forms can be used for a wide variety of intracorporate communications -- bug reports, client contact reports, expense reports, performance reviews, customer complaints, subscription requests and the like.

For communication from one company to another, the issues are more complex; perhaps the best example of widespread intercompany communication is securities trading, interbank wire transfers (see **Release 1.0**, 90-3) and electronic data interchange for buying and selling of goods. The world of EDI is the first market where these issues are being addressed outside financial markets (see our next issue for another example). What's a purchase order? What's a unit? What are the procedures for canceling an order? Delaying delivery? Specifying quality? Some concern transactions, others the goods involved. IBM is working in this direction with industry-specific versions of Systems Application Architecture (for insurance and publishing, so far) that will help define industry standards in computer terms.

Forms may end up handling most routine communication -- call them assisted natural language. They provide the context in which you can enter free text or non-redundant information. Fill in this area, and all the usual terms and conditions apply. Specify two items on this form, because we already know who you are and what you're talking about; we know how to handle such cases! We can interpret your comments in context...

Keep me posted: Groupware and interoperability with "foreign" systems

Aside from communications plumbing and common object protocols and definitions, the HandyWidget must address the problem of staying in synch with a changing outside world -- helping its user keep in touch with other users. As with groupware in general, there are two components: coordination of activities and maintaining the integrity of shared information.

As described in computer terms, maintaining the integrity of shared information is reconciling the user's data set with those of his secretary and the other people he communicates with. An object-oriented database can provide the plumbing to handle the problem, but something higher-level must set the rules of reconciliation in a world of nuance and uncertainty. Does the cozy tone of Juan's April letter invalidate the hostile questions of his March memo? Ivan has a new address; has he got a new job, or has his whole company moved? If he has left, who's available to take over his tasks on the Novokuznetsky project? Which memos would be useful for that new person to see? How does Irina's discovery of a new supplier change our cost assumptions? Where is it recorded and how disseminated that July 22 is a holiday?

The technical underpinnings aren't simply keeping a document database with the ability to link all items on a similar topic (a challenge in itself) but representing the data as objects with rules governing their interaction and

---

1 -- *except for those that begin, "You may not remember me, but..." And if you search the database, maybe you will remember.*

The PS/1 is NeXT?

As a counterpoint, consider IBM's new PS/1. It's a nice system, appropriately positioned at the New York City announcement: Jim Cannavino, resplendent in a sweater, put components one by one into a box -- software, service, support, a modem, on-line software, etc. All the best stuff, added together.

It reminds us of the NeXT machine: The very best of today's technology, packaged together in a single stock-keeping unit for customer convenience. The trouble is, neither machine shows the revolutionary spirit that will create a new market rather than enlarge an existing one. The PS/1, in fact, is the product of years of diligent market research. Prospective customers dutifully supplied their wish lists, and IBM dutifully responded with a system that is not just easy to use but easy to buy. Unfortunately, even now it sports several disparate easy-to-use interfaces: The PS/1's own information service, Prodigy, Microsoft Works, plus whatever the user may happen to use under DOS.

Where IBM and NeXT added up a variety of technologies, GO and SuperScript are attempting to multiply them: "Don't give people what they asked for; give them what they never dreamed of." Of course, this is riskier; your chances of getting it right are much smaller. But if you do get it right you have a whole market to play with, while IBM has to share the existing home computer market with Tandy, Apple, Atari, Amiga and lots of other people who offer some or all of the features on people's wish lists.

Beyond that, the PS/1's signal advantage is that it's a preconfigured communication system -- although it's not being clearly positioned as such. While we're not a big fan of Prodigy, its interface has improved considerably and communication is a primary reason to buy these systems. After you've done your work at home, taught the kids French and played a few games, what else is there? Indeed, we think the big competition for this machine will be home fax, with its single, easy-to-use interface, its simple ready-when-you-are operation and its lack of any pretense at personality.

But the problem is that none of the PS/1's capabilities/applications works fundamentally any better (except for the security of support) as a result of being on this machine. It's the same old ported-software-never-created-a-new-platform problem -- one that the NeXT suffers from too, as we hear more about ported products than about ones built in Objective-C for the NeXT specifically. Yes, users can program on the NeXT with the wonders of object-oriented programming, but no one has yet created the class libraries that will encourage them to do so for commercial applications.

The only way to get such seamless integration is to start fresh. And the only way to do that and make money is to start small. You can't replace the whole world overnight -- which is why no large company is likely to take on the task. Just start off in a specific corner nobody wants such as calendar/communications management and control that, until you can use that base to reach out.

the way they can be modified. (Database transactions are a simplified example of this.) Aside from the focused "groupware" vendors, a number of large companies are working in this area. With luck they may agree on some common objects and thereby multiply the value of their efforts. As Philippe Kahn described it recently -- and as a newly formed Borland research group in Paris will be exploring -- this is the problem of the "asynchronous client" in a client-server system. When you check in at central, what needs updating? The problem is more difficult with rich data types and objects.

In a similar vein, Lotus isn't isolating groupware in Notes; it's working on the concept of group spreadsheets. The answer isn't just storing data in a database server where it can be kept in a database and loaded into a spreadsheet, but keeping assumptions -- relationships among sets of data -- consistent. What can you change, and what is sacrosanct or at least linked? This could also apply to a single user creating a set of scenarios: Which assumptions are linked, and which independent? For example, if the weather is bad, you assume lower traffic overall, but higher sales of umbrellas; yet you may not have represented this explicitly (before now).

Who's on first?

The other part of groupware is work coordination -- delegation, approvals, workflow and other process-oriented coordination that happens more or less independently of the work content. In dealing with others, people make "transactions" beyond purchases and sales (see **Release 1.0**, 88-6). The HandyWidget will have an "understanding" of all of these, but the guy at the other end -- to whom all the work is being delegated, say -- has to share that understanding. What are the rules for confirming an appointment, as determined by the reliability of the person with whom the appointment is scheduled, the importance of the appointment, the prevailing quality of communications? Would Vaclav call us to cancel, or is he the kind who would just not show up? (Ooops -- did he ever confirm in response to our confirmation?) Whose time is most valuable? Who gets priority if there's a conflict? If Irina calls in sick, who handles her urgent messages -- and how are those distinguished from less pressing ones? Who needs to see this before it goes to the boss? Before it goes to the client, the government, the lawyer? The HandyWidget doesn't know -- but it can be told, and follow through reliably.

Object-oriented programming

HandyWidget defines objects globally, not just for a single application -- although of course all the applications have to go along with this scheme. That's the first half. To achieve it, you need a set of agreed-on objects (class libraries) -- people, tasks, resources, orders, all the things your business is about (not just screen or system objects). You also need rules governing their interaction and procedures for manipulating them, as described above. All these high-level notions are ultimately represented as 1s and 0s, but the user never even gets close to that. He needs not an "object-oriented programming language," but an application language he can understand that camouflages the object-oriented language underneath. These capabilities are (sort-of) available right now, just as you could get the capabilities of a spreadsheet with a financial planning language (if you were a programmer or knew one who liked you). In other words, there's the object-oriented language the builder-user uses to build and define objects, and the objects themselves that the HandyWidget end-user can use.

Second half: Calendar programming

Specifically, HandyWidget needs a calendar programming language to get things started: It must know about priorities, schedules, notifications, open items, move-time, move-place, add/remove person, meal/no meal, trip report, duration, travel time/proximity (while I'm in the area, whom else can I see?), maps. It should also have modules for say, expenses, travel arrangements, branches & territories (company-specific), E-mail, fax, priorities, etc. You can't have a useful user language about generalities; but most computer languages are about just that -- computer items such as files, records, procedures, strings. So the HandyWidget's language and its context start out with calendar items and can easily expand to encompass the particulars of any individual's company or business.

It will have capabilities akin to Agenda's, that can manipulate data quickly, provide different views and the like. Agenda started right: It can pick out dates, names, and cross-references, but it is still just matching text strings (aside from dates) and assigning markers to items by the field where the user entered them. Agenda knows about categories and rankings, but it doesn't know about content -- or objects. (Of course, developers can give it some of that capability, as is now happening.) Ultimately, Agenda's can't impose its view of the world on any other applications; its categories and their relationships become meaningless if another application tries to use them. But the advances it made won't be lost.

Likewise, you can get many of HandyWidget's superficial capabilities in a personal manager such as (most elegantly) the new version of Desktop Set from Okna or DayBook from start-up Asymetrix (founded by Microsoft co-founder Paul Allen). DayBook is bundled with Microsoft Windows 3, which alone should ensure its success. But DayBook doesn't go far enough. If you take an item out of the address book and put it in the calendar, it becomes just a name, not a reference to the person in the address list. You can get more flexibility by using ToolBook (DayBook isn't programmable without ToolBook). ToolBook/DayBook won't be the breakthrough we envision because it's not radical enough. It still runs on the same old machine, and doesn't control the user's whole environment.

Finally, you can turn DOS files and associated applications into objects, as NewWave does, but the performance, flexibility and granularity all suffer. Put these all together and you'd get a slow, cumbersome mess usable only by a C++ hacker. Typical DOS applications are too big to load all at one time; you want to be able to get just the functionality you need. (Borland's VROOMM, **Release 1.0**, 89-6, is a step in this direction.) The point is to foster the creation of new applications that can work together, not warmed-over versions of existing DOS ones that can only trade data.

Content and context is the essence of HandyWidget. Sure, you could build a single application loaded with them -- just look at the list on page 8. But people tend to do what tools make it easy to do, and they want more than the capabilities of a single application.

SOME COMPANY SPECIFICS

This is hardly going to be an uncrowded market. GO and Momenta are promising wonderful things, and Apple is rumored to be doing something too. Note Larry Tesler's new job working on "the leading edge of...combining object-oriented programming with miniaturized computing," according to John Sculley as quoted in MacWEEK (15 May, page 10). Sun (under Bill Joy) is working on a new platform, though not necessarily pen-oriented, and Microsoft reportedly has something in the works too: HandyWindows? Since it's based on Windows, we don't see how it can provide the break with the past necessary to create a new market -- as opposed to enlarging the current one, which Windows 3 will do nicely.

The first HandyWidgets will be limited-domain systems, with application-specific capabilities (such as calendar management). But the HandyWidget, unlike the vertical applications built one by one, will be the home base from and through which the user communicates with the rest of the world.

GO Corporation

GO Corporation, with the best-known unannounced notepad computer on the way, is the closest thing to a brand-name start-up since NeXT (for better or worse), with funding of $10 million from Kleiner Perkins, Chancellor Capital; US West, Harvard Management; a star-studded board including chairman John Doerr of KP; Sun co-founder Vinod Khosla; and Metaphor founder Dave Liddle. Most of the top ten software vendors (except Microsoft) have committed to produce software for the GO machine. President and founder is Jerry Kaplan, author of Agenda and a founder of Teknowledge, and author of "Cooperative responses from a portable natural-language query system," his 1979 PhD thesis. Employees include Framework creator Robert Carr and Dan'l Lewin of Apple and NeXT. Overall, GO has an interesting set of influences. They include not just Agenda and ON Technology, but also Liddle's experiences at PARC and at Metaphor with non-standard-hardware systems. (Metaphor's software guts are now being ported to the Intel-OS/2 environment under the guidance of investor IBM.)

There's a certain irony here. GO investor and advisor Mitchell Kapor wanted to create a new operating system at ON Technology (where he's ceo), but he abandoned the effort because he couldn't get support from hardware vendors. It may be GO that achieves this aim by going around the hardware vendors with its own hardware; as Kapor himself said (writing in **Release 1.0**, 88-2), a new platform "can't be created out of what we have today, nor even out of extensions to not-fully-released new systems" [OS/2]. It will take a whole new machine to get that new platform established. There's a second irony in that Kapor and Kaplan were intimately involved in one of the first efforts in this direction -- Agenda, picked up and marketed by Lotus.

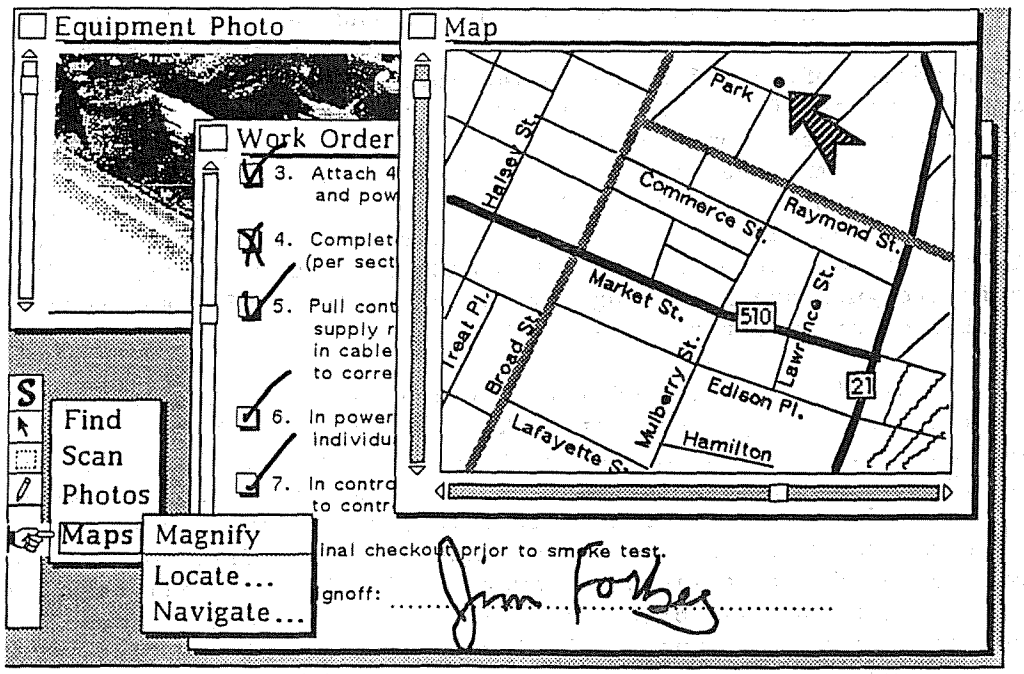
GO is loath to discuss much aside from its pedigree, but it has allowed that it won't be using DOS. Rumor also has it that IBM has taken a serious interest in the company. An operating system based around the GO machine might well provide an interesting successor to DOS some years down the road, while OS/2 continues to be IBM's heavy-duty SAA-compliant environment. Who knows? We're sure such speculations are well beyond any firm plans IBM may have; the future must still unfold. But whatever IBM or GO does, we hope their plans include broad licensing of the GO environment -- emulating the policy that helped make DOS such a success.

SuperScript

Of the vendors with new software platforms, only SuperScript is willing to discuss specifics. Its "Companion" notepad machine, totally nonstandard without even an Intel chip, provides the most promising system by far for development of vertical applications (see below). The company was founded last year by John Kulp, formerly vice president of R&D at Symbolics; its staff of eight (so far) includes four other people from Symbolics and the former product manager for Lotus Agenda. Although SuperScript has no formal connection with Symbolics, its Companion uses Symbolics' Ivory chip and much of its Genera development environment and tools such as Concordia (hypertext; see **Release 1.0**, 88-4) and Statice, an object-oriented database (89-9). It would be ironic if this platform were to be the one on which Symbolics' superb software environment were to become a solid commercial success. "The Ivory chip buys us a lot of power," says Zyg Furmaniuk, "and saves us years in building a rich development environment."

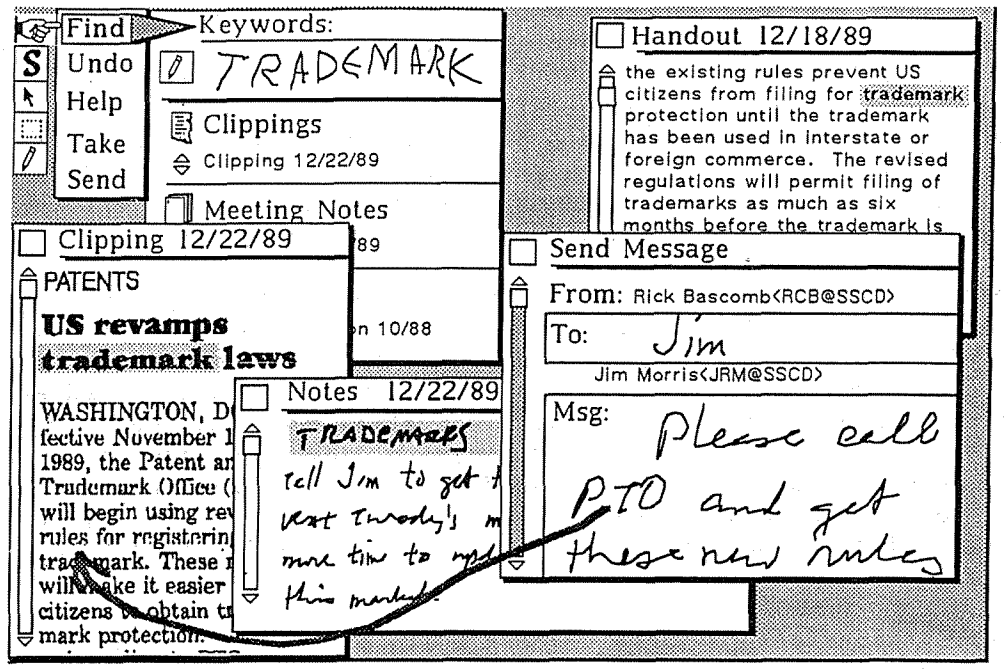
SuperScript's Companion system is being targeted for people building applications for field personnel, but it can also be used in office functions, as shown across. It provides much of what we described above, or at least the

COMPANION SCENARIO: OUTDOOR FIELD REPAIR



Here's a Companion (mock-up) as it might be used by a field-service technician, Jim Forbes. He used the map to find the site, and the photograph to identify the equipment at the site. He checks off his actions as he completes them; the system will warn him if he misses something. If there's a problem (or he can't remember the proper procedures) the system can provide appropriate guidance through hypertext or a diagnostic system. When he signs off on each job, the system records it as completed; later (when he connects by modem, probably), it will update the central office host.

COMPANION SCENARIO: IN-OFFICE INTERACTION



In this mock-up, the Companion is being used by Rick Branscomb, a lawyer. He has done a text search on the word trademark, and selected the two items he wants from a list of matches including clippings, meeting notes and phone messages as well as articles. Last December, the system had automatically indexed the meeting handout, which he had stored digitally; he had read the clipping at left and highlighted the word trademark, which the system retained as a keyword; likewise, he had identified a note to himself as concerning "trademarks." Note that the system classified these last two items by key words, but stored them as (compressed) image, not as text. Now Rick points (with the arrow) to the clipping, and writes a note about it to Jim (whom the Companion recognizes as Jim Morris). The system doesn't read the message to Jim, but simply transmits it to him, along with the clipping and the hypertext link to it. Note that Jim has done all this without disrupting the senior partner's steady drone.

potential for it, with the rich development environment licensed from Symbolics. However, it is far from having the end-user-oriented tools that would be required for the HandyWidget.

The real issue is, how constricting will developers find DOS as they attempt to build applications for the pen computers, and how persuasive (vs. expensive) will SuperScript be? Certainly, it provides far richer facilities for diagnostics, linking text and images, etc., than DOS. ("We have more requests for still video than for voice," says SuperScript's Furmaniuk.) Indeed, the Symbolics environment was made for this kind of application, with its ability to handle objects, and its hypertext and expert systems heritage. But it won't necessarily provide objects at a level convenient for users as opposed to user-builders. Will GO? It will certainly require third-party support as well in either case. But we think GO is more likely to attract the broad, horizontal class libraries that will be necessary.

Momenta

Like GO, Momenta is best-known for its people, but they tend to come out of the hardware rather than the software world. They include Kamran Elahian, founder of CAE Systems and Cirrus Logic, both successful ventures; Shiraz Shivji, who designed the Atari ST; John Fischer, with a high-level background in manufacturing at HP; and former Macintosh marketing manager John Rizzo, also formerly with Weitek. This crew has enabled the company to attract more than $13 million from backers in the US and the Pacific Rim, including Nazem & Co., Institutional Venture Partners, Associated Venture Investors, Sequoia and Walden (backed a US fund with Pacific Rim backing).

Momenta stresses that it's a global company, with offices and product development worldwide, which will enable it to benefit from Japanese hardware developments. The company doesn't seem to have as clear a sense of the software needs as GO does. It seems to be intent on producing super hardware and will do its own environment by default. "We're developing our own complete product at full speed," says Rizzo, "but we'd consider something on DOS or something from GO or someone else if it's good enough. We're doing an interface because there's no one you can go out and buy one from, the way GO seems to be doing hardware because it can't find an existing system to put its software on." They sound fairly complementary to us...

In its marketing materials, Momenta focuses quite heavily on voice recognition (a useful but definitely not universally desired feature). That's even further out than acceptable handwriting recognition, we believe: It won't be realized until there's strong underlying natural-language capability -- probably application/domain-specific for practical reasons. Thus, you can't avoid dealing with applications, user interface (at a functional, not a visual, level) and software. Of course, Momenta may just leave all that to third parties.

Slate

Scottsdale-based Slate Corporation is the only pure software company in the bunch. Slate is working on a variety of applications for pen-based computers, the company says, and leaves it at that. Although it's an independent company funded by Kleiner Perkins and Hummer Winblad Venture Partners, its California offices are in the same Foster City office tower as GO's. Its star-studded staff includes chairman Vern Raburn, also chairman of Symantec and an early employee of both Microsoft and Lotus; president Tom Byers, formerly with Symantec and Digital Research; Dan "VisiCalc" Bricklin, vp of the Boston development center; Mitch Stein, creator of TrueForm; and Dottie Hall, formerly with Symantec, Lotus and Microsoft.

Communication Intelligence and Scenario

Other companies occasionally mentioned in this market include Communication Intelligence Corp., which is licensing both hardware and recognition-oriented technology to undisclosed OEMs, and Scenario Inc., which announced the DynaWriter in May. The DynaWriter is a handwriting-recognition input device, but it's a component rather than a potential platform. Scenario's basic product is a 19-pound DOS-based CD-ROM touchscreen system, great for information publishing but not for the kind of tasks HandyWidget handles.

RESOURCES & PHONE NUMBERS

Phil Hayes, **Carnegie Group**, 1 (412) 642-6900
Jim Dao, **Communication Intelligence Corp.**, 1 (415) 328-1311
Jerry Kaplan, Dan'l Lewin, **GO Corporation**, 1 (415) 345-7400; fax (415) 345-9833
Irene Greif, **Lotus**, 1 (617) 577-8500
John Rizzo, **Momenta**, 1 (415) 969-3876
Constantin Monastyrsky, **Okna**, 1 (201) 460-0677
Judy Bolger (ex-Gold Hill), **Scenario**, 1 (617) 439-6611
Vern Raburn, Dottie Hall, Tom Byers, **Slate Corporation**, 1 (602) 443-7322 or 1 (415) 345-5100
Susan Morgan, **Softview**, 1 (805) 385-5000
John Kulp, Zyg Furmaniuk, **SuperScript**, 1 (617) 876-0787; fax 876-0867
Jay Wurts, **Symbolics**, 1 (617) 221-1000

COMING SOON

- *Making markets: Perestroika in information services; distribution in the USSR.*
- *Trip report: Prague, Warsaw, Sofia, Ljubljana, Zagreb.*
- *Intellectual property laws in Eastern Europe, country by country.*
- *Database by design.*
- *Network navigation.*
- *The Douglas brothers -- Hofstadter and Lenat.*
- *MathSoft and Mathematica.*
- *Application servers.*
- *And much more... (If you know of any good examples of the categories listed above, please let us know.)*

RELEASE 1.0 CALENDAR

July 10-16	***PC World Forum** - Moscow. Sponsored by IDG. An exposition, with a software conference. Call Frank Cutitta, 1 (508) 879-0700, or Karin Griffhorn, 49 (893) 60860.
July 11	**Improving the technology transfer process** - Vancouver, BC. Sponsored by Technology Access Report and Meadowbrook Consulting. Call Michael Odza, (800) 733-1516 or (415) 669-7236.
July 16-18	**Summer computer simulation conference** - Calgary, Alberta, Canada. Sponsored by the Society for Computer Simulation. Contact: William Svrcek, (403) 220-5755, fax (403) 284-4852.
July 23-25	**Sun Expo '90** - San Jose. Sponsored by Publications & Communications. Call Brona Stockton, (512) 331-7761.
July 23-25	**New visions of distribution conference** - Boston. Sponsored by BIS CAP International. Speakers include Portia Isaacson, now at BIS; Mort Rosenthal, Corporate S'ware; David Dukes, Ingram Micro D. Call Karen McMahon, (617) 893-9130.
July 27-29	**ComputerLand's 1990 vendor fair** - Dallas. Contact Dianne Travalini at (415) 734-4160.
July 29-August 3	***AAAI-90** - Boston, in the heart of AI-land East. Sponsored by the American Association for Artificial Intelligence. With speeches by DARPA's controversial Craig Fields; DEC's Sam Fuller. Contact: Claudia Mazzetti, (415) 328-3123.
August 6-7	**International software manager's conference** - Beaver Creek, CO. Sponsored by Viasoft. With Chris Haveman, SPSS; William Atkin, Baker & McKenzie; others. Call Roger Philips, (602) 952-0050.
August 6-10	***SIGGRAPH '90** - Dallas, TX. Sponsored by ACM and IEEE Computer Society's Technical Committee on Computer Graphics. Visualization, virtual reality, more. Call Kathleen Nilles, (312) 644-6610.
August 8-11	**Macworld** - Boston. Sponsor: Macworld. With John Sculley and Jean-Louis Gassee. Call Peggy Kilburn, (617) 361-1472.
August 8-11	**Fifth Annual Snowmass working session** - Snowmass Village, CO. Sponsored by Educational Uses of Information Technology (EUIT), formerly EDUCOM Software Initiative. Keynote: Kenneth "Casey" Green (USC). Call Angela Walsh, (609) 520-3350.
August 13-17	**International parallel processing conference** - St. Charles, IL (25 miles from O'Hare). Sponsored by Pennsylvania State University. Contact: David Padua, (217) 33-4223 or Benjamin Wah, (217) 244-7175, or Roger Anderson, (415) 422-8572.
August 15-17	**Windows conference & exposition** - Boston. What hath Windows 3.0 wrought? Sponsored by Computer Currents. Call Steve Hollis, (415) 601-5000.
September 5-7	***Breakaway 90** - New Orleans. Sponsored by ABCD, for dealers and vendors. With a panel featuring Mike Shabazian, Mike Pickett, Mike Swavely, moderated by Esther Dyson. Contact Jeff Rosenberg, Computer Emporium, (914) 565-6262.
September 9-12	**18th mini/microcomputer industry conference** - Boston, MA. Sponsored by Cowen & Co., new home of Michele Preston and Mary Meeker. Contact: Amy Burns, (617) 523-3221.
September 10-12	**DataStorage 90** - San Jose. Sponsored by Disk/Trend, Inc. and Freeman Associates. The hard-storage industry's main event. Call Darlene Plamondon, (408) 554-6644.
September 10-13	**NetWorld '90** - Dallas. Sponsored by H.A. Bruno. Call Annie Scully or Mark Haviland, (201) 569-8542 or (800) 444-EXPO.

September 11-14	**OpCon East** - Cambridge, MA. Sponsored by Soft*Letter. Focused, practical information for people who manage the real work: manufacturing, marketing, tech support, sales management. Call Tom Stitt, (617) 924-3944.
September 16-20	**Fifteenth annual design management conference** - Edgartown, MA. Sponsored by The Design Management Institute. A neat, multi-disciplinary event. Call Nancy Barry, (617) 236-4165.
September 17-20	**Expert Systems 90** - London, UK. Sponsored by the British Computer Society. Call Fiona Pearson, (011 44) 273 694079.
September 23-25	***Agenda 91** - Laguna Niguel, CA. ...enhanced by a great new place. Sponsored by P.C. Letter/PCW Communications; staged by Stewart Alsop. Call Tracy Beiers, (415) 592-8880.
September 25-27	**International expert systems conference and exposition** - London. Focusing on mainstream concerns: applications, integration, reliability and quality. Sponsored by Learned Information. Contact: Jean Mulligan, 44 (867) 730275.
September 25-27	**PC Expo** - Chicago. Sponsored by H.A. Bruno. Contact: Steve Feher, (201) 569-8542 or (800) 444-EXPO.
September 26-27	***Portable computing & communications** - New York City. Sponsored by IDG. The market gets bigger as the machines get smaller. Call Bonnie MacKeil, (508) 820-8258.
Sept 30-Oct 3	***IIA annual convention/exhibition** - Boston. Sponsored by the Information Industry Association. The gathering in person of all those who communicate on-line and through CD-ROM. Call Linda Cunningham at (202) 639-8262.
October 1-5	**CD-ROM Expo** - Boston. Sponsor: IDG Conference Management. Call Dorothy Ferriter, (800) 225-4698 or (508) 879-6700.
October 3-5	**Seybold Conference** - San Jose. Electronic publishing in all its guises. Call Kevin Howard, (213) 457-5850.
October 3-5	**UNIX Solutions** - Anaheim. A new one, sponsored by the Interface Group. Call Jackie Murphy, (617) 449-6600.
October 3-5	**Knowledge engineering today's marketplace II** - San Francisco. Sponsor: International Association of Knowledge Engineers. With Ed Feigenbaum. Call Julia Ballesteros, (301) 231-7826.
October 7-10	***CSCW '90** - Los Angeles. Computer-supported cooperative work, with a slight (but lessening) academic flavor. Sponsored by ACM. Call Frank Halasz (back at PARC after a tour at MCC) at (415) 494-4750, or Tora Bikson, (213) 393-0411.
October 8-12	**SofTool USSR** - Moscow. Sponsors: All-Union Society of Information & Computer Technology, Phargo Group. Software tools and technology. Call Gene Zarwell, 1 (703) 284-7330, or Levan Tatunashvili, 7 (883-2) 99-94-03.
October 8-12	**Interop '90** - San Jose. Sponsored by Advanced Computing Environments. Call Dan Lynch, (415) 941-3399.
October 10-12	**INFO '90** - New York City. Sponsored by Cahners Exposition Group. Call Marilyn Harrington, (203) 352-8477.
October 10-13	***Software Publishers Association conference** - New Orleans. Call Ken Wasch, (202) 452-1600.
October 14-16	***ETRE: European Technology Roundtable Exhibition** - Opio (near Nice), France. Co-sponsor: Le Monde. "Technology and the new Europe." With a panel by Esther Dyson on Eastern Europe, speeches by luminaries. Call Alex Vieux, (415) 321-5544.
October 17	**New York PC User Group meeting** - New York City. Jim Manzi, Lotus. Call John McMullen at (914) 245-2734.
October 18	**Massachusetts Computer Software Council's fall membership meeting** - Boston. Call Joyce Plotkin at (617) 437-0600.

October 18-19	**The world of object-oriented applications** - Amsterdam, The Netherlands. Sponsored by Object Management Group and Patricia Seybold's Office Computing Group. Call Jeffrey Perfect, (617) 742-5200.
October 21-24	****EDventure East-West High-Tech Forum** - Budapest, Hungary. By popular demand. Explore the problems and opportunities of high-tech business in Eastern Europe and meet your peers in a limited-attendance conference focused on contacts, not speeches. Sponsored by EDventure Holdings, with a roster of speakers and attendees from both sides. Call Daphne Kis, (212) 758-3434. By invitation only.
October 21-25	**OOPSLA/ECOOP 90** - Ottawa, Canada. Joint conference on object-oriented programming: Systems, languages and applications. Sponsored by ACM. Call David Thomas, (613) 228-3535.
October 22-24	**Electronic Messaging '90** - San Francisco. Sponsor: Electronic Mail Association. Call Anne Spence, (703) 522-7111.
Oct 31-Nov 2	**UNIX Expo** - New York City. Sponsored by National Expositions. Contact: Don Berey, (212) 391-9111.
November 4-7	***ADAPSO management conference** - Phoenix. Contact: Ellen Kokolakis at (703) 522-5055.
November 5-9	**ICCC 90** - New Delhi, India. Hosts: Government of India Departments of Electronics and Telecommunications. Sponsored by The International Council for Computer Communication. Call P. Gupta, 91 (11) 683-0087, or fax, 91 (11) 684-4652.
November 6-7	**1990 ACM conference on critical issues** - Arlington, VA. "Managing complexity and modeling reality." Call Jim Adams, (212) 869-7440.
November 6-9	**Software Development Conference** - Boston. Sponsor: Miller Freeman Publications. Call KoAnn Tingley, (415) 995-2472.
November 12-16	***Comdex Classic** - Las Vegas. The real thing. Sponsored by Interface Group. Call Elizabeth Moody at (617) 449-6600.
November 12-16	**Neuro-Nimes '90** - Nimes, France. Neural networks and their applications. Sponsor: Association for Cognitive Research. Contact: Annick Crozet, (331) 47806452 or (331) 47807000.
November 14-16	**Pacific Rim international conference on artificial intelligence '90** - Nagoya, Japan. Sponsored by the Japanese Society for Artificial Intelligence. Special sessions for AI in engineering and AI and large-scale information. Call Shigero Sato at (813) 479-5535 or fax (813) 479-7433.
December 3-4	***Emerging Technologies conference** - Innisbrook, FL. For developers eager to use the latest in technology -- object-oriented programming, cooperative processing, IBM repository, notepad computing, image management and the like. Sponsored by the Technology Information Services Committee of ADAPSO's Software Industry Division. Contact: Phyllis Cockerham, (703) 522-5055.
December 3-7	**Toulouse '90** - Toulouse, France. Software engineering and its applications. Sponsored by EC2. Contact: Jean-Claude Rault, (331) 47807000 or fax (331) 47806629.
December 5-8	***CASE '90** - Irvine, CA. The fourth international workshop on computer-aided software engineering. Sponsored by Index Technology, IEEE and several academic insititutions. Call Ron Norman, (619) 594-3734.
December 10-11	**Virtual reality: Theory, practice and promise** - San Francisco. Sponsored by Multimedia Review. Call Marilyn Reed, (800) 635-5537 or (203) 226-6967.

1991

January 21-24	**UniForum '91** - Dallas. Sponsored by UniForum Association. Call Ed Palmer, (408) 986-8840.
February 12-15	***Software Development '91** - Santa Clara. Sponsored by Miller Freeman. Call Lynne Mariani or Angela Hoyte, (415) 995-2471.
February 24-28	**CAIA-91** - Miami. Sponsored by IEEE Computer Society. Call Jeff Pepper, (412) 642-6900.
March 3-7	***Seybold Seminars '91** - Boston. Call Kevin Howard, (213) 457-5850.
March 10-13	****EDventure Holdings PC (Platforms for Computing) Forum** - Tucson, AZ (again). Sponsored by us! Contact: Daphne Kis, (212) 758-3434.
March 12-15	**OpCon West** - Palo Alto, CA. Sponsored by Soft*Letter. Call Tom Stitt, (617) 924-3944.
March 13-20	**Hannover Fair CeBIT '91** - Hannover, West Germany. Sponsor: Hannover Fairs USA. Call Donna P. Hyland, (609) 987-1202.
March 17-20	***Software Publishers Association spring symposium** - San Francisco. Call Ken Wasch, (202) 452-1620.
September 11-14	***Software Publishers Association annual conference** - Orlando. Call Ken Wasch, (202) 452-1620.

Please let us know about any other events we should include. -- Denise DuBois

**The asterisks indicate events we plan to attend. Lack of an asterisk is no indication of lack of merit.*

Release 1.0 is published 12 times a year by EDventure Holdings, 375 Park Ave., New York, NY 10152; (212) 758-3434. It covers pcs, software, CASE, groupware, text management, connectivity, artificial intelligence, intellectual property law, and emerging technology markets in Central Europe and the Soviet Union. Editor & publisher: Esther Dyson; associate publisher: Daphne Kis; circulation & fulfillment manager: Lori Mariani; executive secretary: Denise DuBois; editorial consultant and copy chief: William M. Kutik. *Copyright 1990, EDventure Holdings Inc. All rights reserved. No material in this publication may be reproduced without written permission; however, we gladly arrange for reprints or bulk purchases.* *Subscriptions cost $495 per year, $575 overseas.*

SUBSCRIPTION FORM

Please enter my subscription to **Release 1.0** at the rate of $495 per year in the U.S. and Canada. Overseas subscriptions are $575, airmail postage included. Payment must be enclosed. Multiple-copy rates on request. Satisfaction guaranteed or your money back.

Name _____

Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Telephone _____

How did you hear about Release 1.0? _____

_____

SIGGRAPH90 90-6

Please fill in the information above
and send with your check payable to:

EDventure Holdings Inc.
375 Park Avenue, Suite 2503
New York, NY 10152

If you have any questions, please call us at (212) 758-3434.

Daphne Kis
Associate Publisher